

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting a data object having an identifier (ID) from the first storage location;

determining whether an identifier (ID) of the data object is stored in a transactional lock object;

when the ID is not stored in the transactional lock object, storing the ID in a second the transactional lock object;

determining whether the ID is stored successfully in the second a permanent lock object; and;

when the ID is not stored in the permanent lock object, upon a successful storage, locking the data object in the first storage location by storing the ID in a first the permanent lock object, thereby indicating that the data object is stored at the first storage location;

deleting the ID from the second transactional lock object after storing the ID has been stored in the first permanent lock object;

storing the data object at the second storage location;

deleting the data object from the first storage location; and

deleting the ID from the first permanent lock object, thereby indicating that the data object is not stored at the first storage location, after deleting the data object has been deleted from the first storage location.

2. (Previously Presented) The method of claim 1, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

3. (Currently Amended) The method of claim 1, wherein the data object is stored in a file and wherein an assignment of the ID to the file or to a name of the file is stored in the first permanent lock object.

4. (Currently Amended) The method of claim 1, wherein the first permanent lock object is stored on a nonvolatile storage means.

5. (Currently Amended) The method of claim 1, wherein the ID is stored in the second transactional lock object after selecting the data object from the first storage location.

6. (Currently Amended) The method of claim 1, wherein the ID is stored in the second transactional lock object before storing the data object is stored at the second storage location.

7. (Currently Amended) The method of claim 1, wherein storing the ID in the first permanent lock object further comprises:

storing IDs of other data objects in the first permanent lock object before storing the data object at the second storage location.

8. (Currently Amended) The method of claim 1, further comprising:

~~when checking, before storing the ID in the first lock object, whether the ID is has-~~
been stored in the first permanent lock object, ~~and if the ID has been stored,~~ skipping
storing the data object at the second storage location.

9. (Currently Amended) The method of claim 1, further comprising:

determining ~~checking~~ whether the data object is ~~contained~~ stored in the second
storage location[.]; and

when ~~[[if]]~~ the data object is ~~contained~~ stored in the second storage location,
skipping storing the data object at the second storage location.

10. (Currently Amended) The method of claim 9, wherein ~~the checking is-~~
~~performed by querying~~ determining whether the data object is stored in the second
storage location comprises determining whether the ID is stored in the first permanent
lock object.

11. (Currently Amended) The method of claim 1, further comprising:

determining ~~checking~~ whether storing the data object ~~has been stored~~ in the second storage location ~~[[,]]~~ was successful; and ~~[[if]]~~

when storing the data object ~~has not been stored in the second storage location~~
was not successful, skipping deleting the data object from the first storage location and skipping deleting the ID from the first permanent lock object.

12. (Previously Presented) The method of claim 1 for use in an enterprise resource planning software.

13. (Currently Amended) A computer system for processing data, the computer system comprising:

memory means for storing program instructions;

input means for entering the data;

storage means for storing the data;

a processor responsive to the program instructions, wherein the program instructions comprise program code means for performing a method for moving data objects in the computer system from a first storage location to a second storage location, the method comprising:

selecting a data object ~~having an identifier (ID)~~ from the first storage location;

determining whether an identifier (ID) of the data object is stored in a transactional lock object;

~~when the ID is not stored in the transactional lock object, storing the ID in a second the transactional lock object;~~

~~determining whether the ID is stored successfully, and in a permanent lock object;~~

~~when the ID is not stored in the permanent lock object, upon a successful storage, locking the data object in the first storage location by storing the ID in a first the permanent lock object, thereby indicating that the data object is stored at the first storage location;~~

~~deleting the ID from the second transactional lock object after storing the ID has been stored in the first permanent lock object;~~

~~storing the data object at the second storage location;~~

~~deleting the data object from the first storage location; and~~

~~deleting the ID from the first permanent lock object, thereby indicating that the data object is not stored at the first storage location; after deleting the data object has been deleted from the first storage location.~~

14. (Canceled).

15. (Currently Amended) A computer readable storage medium comprising instructions for performing a method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting a data object having an identifier (ID) from the first storage location;

determining whether an identifier (ID) of the data object is stored in a transactional lock object;

when the ID is not stored in the transactional lock object, storing the ID in a second the transactional lock object;

determining whether the ID is stored successfully, and in a permanent lock object;

when the ID is not stored in the permanent lock object, upon a successful storage, locking the data object in the first storage location by storing the ID in a first the permanent lock object, thereby indicating that the data object is stored at the first storage location;

deleting the ID from the second transactional lock object after storing the ID has been stored in the first permanent lock object;

storing the data object at the second storage location;

deleting the data object from the first storage location; and

deleting the ID from the first permanent lock object, thereby indicating that the data object is not stored at the first storage location, after deleting the data object has been deleted from the first storage location.

16. (Canceled).

17. (Previously Presented) The computer readable storage medium of claim 15, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

18. (Currently Amended) The computer readable storage medium of claim 15, wherein the data object is stored in a file and wherein an assignment of the ID to the file or to a name of the file is stored in the first permanent lock object.

19. (Currently Amended) The computer readable storage medium of claim 15, wherein the first permanent lock object is stored on a nonvolatile storage means.

20. (Currently Amended) The computer readable storage medium of claim 15, wherein the ID is stored in the second transactional lock object after selecting the data object from the first storage location.

21. (Currently Amended) The computer readable storage medium of claim 15, wherein the ID is stored in the second transactional lock object before storing the data object ~~is stored~~ at the second storage location.

22. (Currently Amended) The computer readable storage medium of claim 15, wherein storing the ID in the first permanent lock object further comprises:
storing IDs of other data objects in the first permanent lock object before storing the data object at the second storage location.

23. (Currently Amended) The computer readable storage medium of claim 15, wherein the method further comprises:

~~when checking, before storing the ID in the first lock object, whether the ID is has-~~
been stored in the first permanent lock object, ~~and if the ID has been stored,~~ skipping
storing the data object at the second storage location.

24. (Currently Amended) The computer readable storage medium of claim
15, wherein the method further comprises:

~~determining~~ checking whether the data object is ~~contained~~ stored in the second
storage location~~[[,]]~~ and

~~when~~ ~~[[if]]~~ the data object is ~~contained~~ stored in the second storage location,
skipping storing the data object at the second storage location.

25. (Currently Amended) The computer readable storage medium of claim
24, wherein ~~the checking is performed by querying~~ determining whether the data object
is stored in the second storage location comprises determining whether the ID is stored
in the first permanent lock object.

26. (Currently Amended) The computer readable storage medium of claim
15, wherein the method further comprises:

~~determining~~ checking whether storing the data object has been stored in the
second storage location~~[[,]]~~ was successful; and ~~[[if]]~~

~~when storing~~ the data object ~~has not been stored~~ in the second storage location
was not successful, skipping deleting the data object from the first storage location and
skipping deleting the ID from the first permanent lock object.

27. (Previously Presented) The computer readable storage medium of claim 15, wherein the computer readable medium is provided as part of a computer program product.

28. (Currently Amended) A computerized system for ~~processing~~ moving data objects from a first storage location to a second storage location, the computerized system comprising:

a processor executing program instructions;

means for selecting a data object ~~having an identifier (ID)~~ from the first storage location;

means for determining whether an identifier (ID) of the data object is stored in a transactional lock object;

means for storing the ID in a ~~second~~ the transactional lock object when the ID is not stored in the transactional lock object;

means for determining whether the ID is stored ~~successfully, and upon a successful storage, locking the data object in the first storage location by~~ in a permanent lock object;

means for storing the ID in a ~~first~~ the permanent lock object, ~~thereby indicating that the data object is stored at the first storage location~~ when the ID is not stored in the permanent lock object;

means for deleting the ID from the ~~second~~ transactional lock object after storing the ID ~~has been stored~~ in the first permanent lock object;

means for storing the data object at the second storage location;

means for deleting the data object from the first storage location; and

means for deleting the ID from the first permanent lock object, ~~thereby indicating that the data object is not stored at the first storage location~~, after deleting the data object ~~has been deleted~~ from the first storage location.

29. (Previously Presented) The computer system of claim 28, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

30. (Currently Amended) The computer system of claim 28, wherein the data object is stored in a file and wherein an assignment of the ID to the file or to a name of the file is stored in the first permanent lock object.

31. (Currently Amended) The computer system of claim 28, wherein the first permanent lock object is stored on a nonvolatile storage means.

32. (Currently Amended) The computer system of claim 28, wherein the ID is stored in the second transactional lock object after selecting the data object from the first storage location.

33. (Currently Amended) The computer system of claim 28, wherein the ID is stored in the ~~second~~ transactional lock object before storing the data object is stored at the second storage location.

34. (Currently Amended) The computer system of claim 28, wherein the means for storing the ID in a first the permanent lock object further comprises:
means for storing IDs of other data objects in the first permanent lock object before storing the data object at the second storage location.

35. (Currently Amended) The computer system of claim 28, further comprising:
~~means for checking, before storing the ID in the first lock object, whether the ID has been stored in the first lock object, and if the ID has been stored,~~ skipping storing the data object at the second storage location when the ID is stored in the permanent lock object.

36. (Currently Amended) The computer system of claim 28, further comprising:
means for checking determining whether the data object is contained stored in the second storage location~~[[,]]~~; and
means for if the data object is contained in the second storage location, skipping storing the data object at the second storage location when the data object is stored in the second storage location.

37. (Currently Amended) The computer system of claim 36, wherein the means for ~~checking~~ determining whether the data object is stored in the second storage location comprises means for ~~querying~~ determining whether the ID is stored in the first permanent lock object.

38. (Currently Amended) The computer system of claim 28, further comprising:

means for ~~checking~~ determining whether storing the data object ~~has been stored~~ in the second storage location was successful; and

means for ~~skipping, if the data object has not been stored,~~ deleting the data object from the first storage location and ~~skipping deleting the ID from the first permanent lock object~~ when storing the data object in the second storage location was not successful.

39. (Currently Amended) The method of claim 1, wherein storing the ID in the second transactional lock object indicates that an action is being performed on the data object.

40. (Currently Amended) The method of claim 39, wherein deleting the ID from the second transactional lock object indicates that the action is not being performed on the data object.

41. (Currently Amended) The computer system of claim 13, wherein storing the ID in the ~~second~~ transactional lock object indicates that an action is being performed on the data object.

42. (Currently Amended) The computer system of claim 41, wherein deleting the ID from the ~~second~~ transactional lock object indicates that the action is not being performed on the data object.

43. (Currently Amended) The computer readable storage medium of claim 15, wherein storing the ID in the ~~second~~ transactional lock object indicates that an action is being performed on the data object.

44. (Currently Amended) The computer readable storage medium of claim 43, wherein deleting the ID from the ~~second~~ transactional lock object indicates that the action is not being performed on the data object.

45. (Currently Amended) The computerized system of claim 28, wherein storing the ID in the ~~second~~ transactional lock object indicates that an action is being performed on the data object.

46. (Currently Amended) The computerized system of claim 45, wherein deleting the ID from the ~~second~~ transactional lock object indicates that the action is not being performed on the data object.